EasyChair Preprint

№ 8267

Deep Structural Estimation for Non-Linear Distortion Correction

Daniel Cyrus, Jungong Han and David Hunter

June 12, 2022

Volume 41 (2022), Number 8 pp. 1–6 COMPUTER GRAPHICS *forum*

Deep Structural Estimation for Non-linear Distortion Correction

D. Cyrus J. Han D. Hunter

Prifysgol Aberystywth University.
Cymru, DU. Wales, UK

Abstract

We propose a practical novel method to correct non-linear distortions in videos and single images, which we train a convolutional neural network (CNN) to recognize multiple distortions by estimating image structure. We first employed a VGG16 model to extract features to retain substantial pixels from input images. We designed a CNN, trained by annotated dataset to predict a window frame that visually defined the distortion. A drawing model uses network outputs to generate a grid fitting the window frame. The grid deforms to the corrected sample to render the final image. We use headless rendering mode to enhance correction speed and efficiency. Finally, the experimental results demonstrate that our algorithm outperforms other methods on both time assumption and accuracy. (see https://github.com/danielcyrus/DistortionCorrection)

1. Introduction

Distortion is a major problem in digital images which originates when the field of view is evenly mapped onto the image sensor. These distortions alter the visual characteristics of the image, altering the relative positions of objects in the image and causing straight lines to become curved. Correcting these distortions is necessary for many useful image analyses.

Standard calibration techniques use a physical target with known properties to estimate parameters of a lens model (e.g. [LLC*21, HZZG20]). This requires physical access to the camera and markers with known properties (e.g., chess board or dot board) to estimate the parameters of a lens distortion function. In situations where physical access is impractical, distortion correction needs to rely solely on the visual properties of the image. For example an urban environment contains a large number of straight edges, if we find consistent curves in a image of an urban environment then maybe we straighten these curves to produce an undistorted image. These properties include vanishing points, co-planar circles and repeated textures [HGSE*18, ABAO17, YZW20]. These methods rely on accurate detection on these features, however these features can easily be obscured by other objects in the scene, by low sample resolution or by similarity with surrounding features. More recently deep-learning techniques, such as Convolutional Neural Networks (CNNs) have been employed to overcome these limitations [LZSL19]. CNNs can combine information from across the image and so are less vulnerable to obscuration of image features.
In this paper, we propose an algorithm for blind distortion correction on single images using a CNN to estimate parameters of a two-dimensional polynomial mapping. Results of applying our algorithm are given for both synthetic and photographic images. Figure 1 shows a sample of our result of how our method can apply correction on distorted areas.

Figure 1: Original image(Right) and applied correction resulted from our method(Left)

2. Related Work

Distortion correction has been widely studied which has a significant impact in various applications (e.g., broadcasting videos, scientific images and images with industrial applications). These algorithms consist of two components; a parameterised model of the distortion and an algorithm for finding the appropriate parameters for a given camera. The models under consideration generally apply for images from wide angle camera lenses. These images suffer from known distortion such as projective and radial distortion.
For single images with unknown camera information, estimating the distortion mostly relies on vanishing points, coplanar circles and repeated textures [HGSE*18]. despite their success in correcting single type of distortion, these approaches are not applicable to high definition wide images and multiple distortions in one image.

Blind Geometric Distortion Correction is one possible approach to address this limitation [LZSL19], which is essentially a two-step approach. In the first step, synthetic dataset is used to compare the outputs with the original images to estimate the distortion, in the second step, several iterations are required to detect each distortion and resample the images. Alongside multiple distortions we were inspired by the research determining a grid from images by Tian et al [TN11]. However, the work is limited to text-based image distortion to perform the optical character recognition (OCR), the grid can be estimated in a 3d scene with acceptable accuracy.

2.0.0.1. Single projective distortion. The primary basis of blind calibration is finding features such as edges and making assumptions about the content of the image. Previous author have employed Hough transform [AFAGSC14], Canny edge detection [RLZS14] and arc detection [PH95] to extract features. These methods are highly focused on image detail and can struggle when these details are vague or obscured. More recent methods have employed deep learning techniques to combine information from a wider range of features and from larger areas of the image. CTRL-C [LGL*21] used a Transformer neural model that combines both raw image information with a line segment detection algorithm for estimating parameters of a standard lens model. A similar method for projective distortion is to parse wireframes with detecting straight lines [HWZ*18]. These methods are tailored for urban environments.

2.0.0.2. Single radial distortion. Circular arc detection used in [BD13] to estimate camera parameters. Circular arc are particularly suitable for estimation of radial distortions. López Antequera et al. [LMG*19] proposed a parameterization for radial distortion: they estimate distortion offset and roll angle with a proposed loss function based on point projection. DeepCalib [BERB18] works on wide field-of-view (FOV) cameras with 3d reconstruction methods, which only works for 180° camera lenses. Another similar method is [WY21] with single purpose for fisheye distortion correction, they achieved a large distortion correction with over 180° projection of FOV. Use of line segments is not limited to standard lens models, Vijay et al. [KKL*20] used a reduced set of line segments chosen for there importance in object modelling to perform an accurate estimation for wide-angle cameras but limited to fisheye distortion.

2.0.0.3. Multiple distortions. Most algorithms are bespoke models mostly designed for a limited set of distortion types, Li et al. [LZSL19] trained a CNN to classify multiple distortions; barrel distortion, pincushion, rotation, shear, perspective and wave distortion. In addition, they proposed A new resampling method with faster convergence, the method should run iteratively for each distortion.

2.0.0.4. Realworld measurement using calibrated camera. Many applications require estimation of both camera location and distortion properties. These algorithms often focus on feature tracking as these features can be used both for estimation focal properties of lens and estimation of camera motion (e.g. [GCH*02]). In cases where multiple views of the same scene are available features in multiple views can be reconciled to improve model accuracy [LN18,JAC*21]. Sports camera calibration can take advantage of the regular layout and patterns of professional sporting arenas. Chen et al. fitted images of a soccar pitch to a known template using edge detection [CL19]. Sha et al. used CNNs to implement segmentation and fitted a known template to segmented areas. [SHF*20].

3. Method

The primary basis of calibration is finding features by encoding and decoding objects. Previous authors used Hough transform [AFAGSC14], deep Haugh transform [ZHZ*21], Canny edge detection [RLZS14], arc detection [PH95] to extract features. As they are basic methods to extract image features and possible approaches to estimate image orientation, distance to the origin and also distortion. Since basic methods eradicate image fundamental features we removed traditional methods and replaced them with a VGG network to retain all obligatory pixels. Therefore, more pixels in the same direction are observed as lines or curves, and then proportional end-to-end polynomial lines are drawn over pixels. This technique highlights distortion throughout the image. We have observed that many images contain horizontal and vertical straight line elements such as walls, windows, horizon lines etc. that have been curved by lens distortion. Our method attempts detect and correct these features. Human observers identified and located these lines within a set of distorted images. The images were annotated with four polynomial curves; two horizontal near the top and bottom of the image, and two vertical near the left and right of the images. Each curve is placed on an identifiable feature that the annotator believes would be straight lines in an undistorted image. The parameters of these polynomials were used to train the outputs of the CNN. We first annotate input images by drawing four curves, then third order coefficients are calculated from each curve. Basically, coefficients are decimal values which are rescaled into a 0 to 1 range to be validated in our network. Coefficients are utilized to estimate the shape of the distortion grid. In general, our network feeds from input images and is trained by coefficients.

3.1. Distortion Grid

We have observed that many images contain horizontal and vertical straight line elements such as walls, windows, horizon lines etc. that have been curved by lens distortion. Our method attempts detect and correct these features. Human observers identified and located these lines within a set of distorted images. The images were annotated with four polynomial curves; two horizontal near the top and bottom of the image, and two vertical near the left and right of the images. Each curve is placed on an identifiable feature that the annotator believes would be straight lines in an undistorted image.

We wanted an algorithm that is capable of blind calibration over a wide variety of camera types without making assumptions about the lens. In order to train our neural network we need a function that is parameterisable, with a limited number of parameters to avoid over-fitting and roughly orthogonal parameters as this will make training easier. We chose a polynomial curves as these were flexible enough to describe the image features with relatively few parameters.

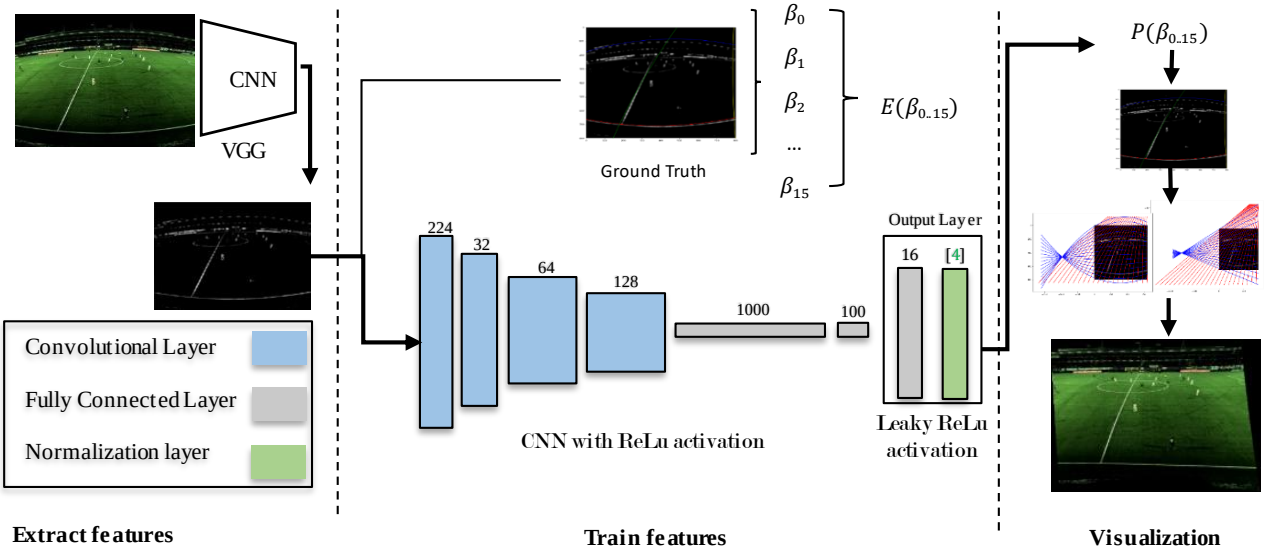

Figure 2: Overview our network structure and whole algorithm process. Ground Truth represent expected coefficients. VGG output represent train images. Blue and red guidelines on right images illustrate the primary structure of grids calculated from predicted coefficients.

The algorithm assumes that the annotated curves should be straight lines and applies a warp to straighten the lines. The warp function models the distortion as if the image was embedded on a flat surface and then bent using a two-dimensional polynomial function (see fig 3). In our algorithm this polynomial was approximated using a piece-wise linear grid. The shape of the polynomial was anchored using the four annotated curves. Thus the two-dimensional polynomial provides a smoothing function to interpolate between the annotated lines over the rest of the image. A mapping was applied to each grid intersection to calculate its position in the straightened configuration. Used OpenGL to render the final straightened image. Through the geometric spatial estimation [CFO93], curve intersections are detected and stored in a matrix (see Figure 2. Blue and red guide lines) to define each square corner. The matrix specifies into a vertex buffer by OpenGL, each vertex match to a quadrilateral primitive 3D shape and layout the grid model, same process take place for straight grid.

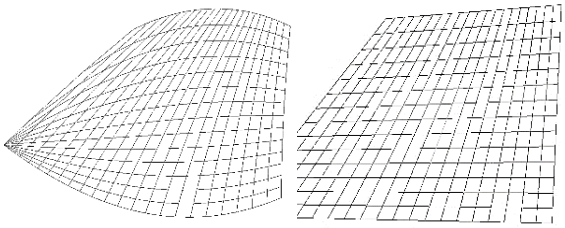

Figure 3: Estimated image grid (left), corrected final grid shape (right).

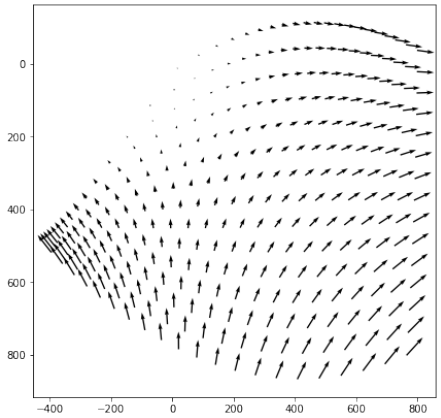

Figure 4: Visualization of pixel movement, direction and distance, estimated from reference and target grid.

3.2. Network Architecture

Our algorithm uses a modified VGG16 [SZ15] Convolutional Neural Network to learn the parameters of four hand-annotated third-order polynomial curves for each image. Our model maintains all but the final layers of the VGG16 neural network, which extracts image features. The output layer consists of a fully connected layer with 16 nodes activated by Leaky ReLU. The outputs 16 nodes correspond to the 16 parameters of the 4 polynomial lines.

The raw parameters of the polynomical contain values outside the ranges generally supported by Leaky ReLU, a normalisation step is required to map the values into an appropriate range. fol-

lowed by a normalization layer, to match the output with rescaled coefficients. All activations (outputs) a in a layer are normalised as $\hat{a}$:

$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l \quad \sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_i^l - \mu^l)^2} \tag{1}$$

Where H denotes the number of hidden units in a layer. σ and μ denotes the normalization term but with different training cases.

3.3. Dataset structure

Our dataset consisted of 4500 photographic images from the places dataset, each image scaled and converted to grey-scale to fit requirements of the VGG16 network. For the training set 4200 images were manually annotated by placing markers along four edge-like features in the image and fitted four polynomial curves using least-squares regression. The edge-like features were chosen such that one was horizontal and located near the top of the image, one was horizontal and located near the bottom of the image and two vertical features near the left and right of the images respectively. We first use original images from places dataset [ZLK*17], then we include manipulated images from [LZSL19] dataset such as barrel, wave and pincushion. Thus, our training dataset includes a wide variety of different distortion types.
We expanded the data set by creating synthetic images based low distortion manipulations of our manually annotated set. Images and their annotations were artificially rotated. The ultimate file saves normalized coefficient to a standard 0∼1 value. Overall, the dataset contains Original images, Gray images, data frame normalized point files.

3.4. Resampling

A part from time-assumption and accuracy, we aimed for reducing mean square error (MSE) [WB09] on the target corrected image. Moreover, manipulate correction accomplished in a single iteration from reference grid to target by linear interpolation. First, the distorted image perches on the estimated grid, then all coordinates move to a new position given from the straight grid. The final deformed grid remains unchanged for all frames within video correction. Figure 4 shows the manner of pixel transformation.

4. Experiments

In this section we report our results. We first analyze our network predictions in section 4.1 and then we discuss our corrected results. In section 4.2, we compare our results with other methods, then a comparison is proposed to evaluate the accuracy and functionality of our method. Section 4.3 shows how we benefit from headless mode for fast resampling in image correction.

4.1. Network Prediction

Our proposed network generates sixteen normalized values which stride model fitting to 4, third-order regression. After denormalization we compare the interval between the predicted outputs and ground truth on our test dataset to estimate the network accuracy, see figure 7. However, current precision is admissible, expansion dataset causes the network to yield better.

4.2. Evaluation

We first use original images from places dataset [ZLK*17], then we include manipulated images from [LZSL19] dataset such as barrel, wave and pincushion. Thus, our train dataset is included both original and distorted images.
To evaluate the performance of our network, a structural similarity (SSIM) method and mean square error (MSE) [WB09, WBSS04] is used to compare with original images, see equation 2.

$$\begin{aligned} SSIM(x,y) &= [l(x,y)^{\alpha}.c(x,y)^{\beta}.s(x,y)^{\lambda}] \\ MSE &= \frac{1}{n}\Sigma_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \end{aligned} \tag{2}$$

However, the aim of this research is mainly to focus on non-linear distortion a fair linear comparison utilizes to appraise with other methods, see Figure 6.

4.3. Headless mode resampling

We perform a high performance resampling strategy that relies on headless rendering to reduce pixel movement iteration. The headless mode enables vertices to render in backend with graphics accelerators. First, a buffer vertex fit the distorted image grids that were extracted from video frames, see Figure 3. The buffer render with new vertices positions of corrected gird shape. Finally, rendered buffers are saved into video frames. Our approach has been implemented on a machine with NVIDIA Tesla k80, Intel(R) Xeon(R) CPU 2.30GHz and 13GB of RAM. Table 2 shows our technique efficiency on different video size through 200 frames:

Resolution	[LZSL19] $\overline{u}$	Ours $\overline{u}$
256×256	38173.08	**273.2222**
1920×1080	-	**7541.548**
4450×2000	-	**26489.16**

Table 2: Time efficiency on different frame size, each column represents width and height of the video frames.

Where $\overline{u}$ denotes average time efficiency in microsecond for images with minimum 72dpi. The values in table 2 obviously shows our method benefits from headless mode as it requires one iteration for all distortions through the resembling.

5. Discussion

In this paper we present a state-of-the-art technique to enhance multiple corrections, nevertheless, there are always limitations. Basically, the algorithm needs identifiable lines and curves to wield the correction, it poorly operates if there are none. For images with less features such as sky and sea, images with high dens background such as contradictory curves closed to each other, estimating a correction cannot be represented by our model. On the other hand, curvy structures may be detected as distortion, for example metal circular structures, non-flat walls and curved tree trunks. Thus, our model may change the nature of non-distorted images. Lastly, the robustness of our model could be enhanced with more

Figure 5: Example of distortion correction for ultra-wide real-time video. Corrected video (left) and original Video (right).

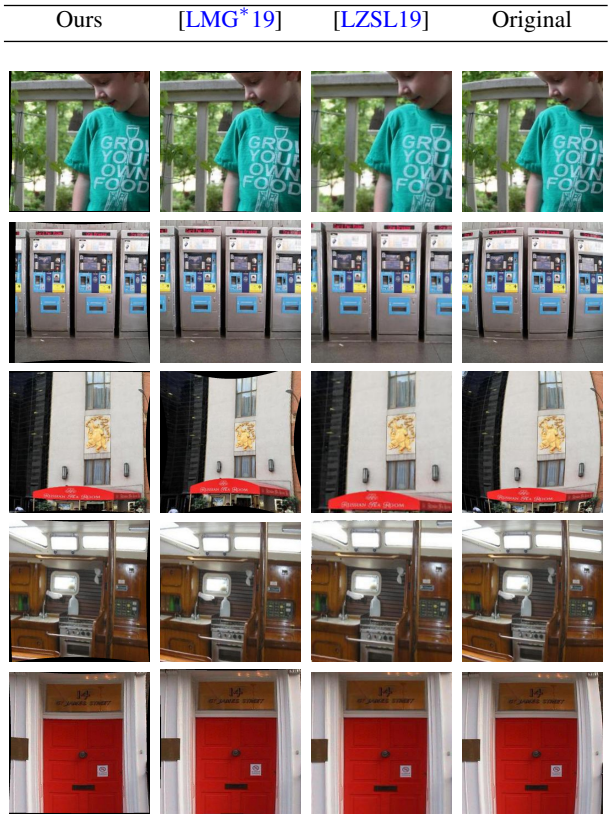

Figure 6: Qualitative comparison of our results with Deep Calibration [LMG*19] and Blind geometric correction [LZSL19].

Model	SSIM	MSE
Ours	**0.412**	**1227.77**
[LMG*19]	0.360	1640.42
[LZSL19]	0.411	1279.36

Table 1: Structural similarity(SSIM) and Mean Square Error(MSE) statistics of our approach using 100 images.

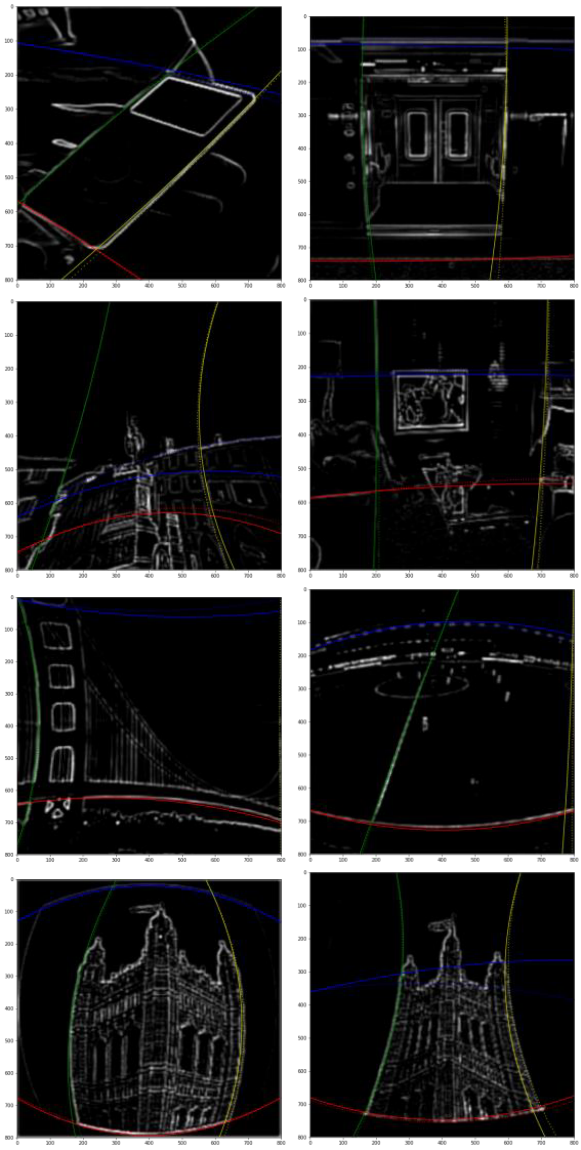

Figure 7: Example result of comparison of our network predictions (solid lines) and ground truth (dot cross lines).

image transformation.
It also needs to be considered that we compared our method with other researches which were published in 2019, for this reason, they implemented their methods for a wide variety of images. Although recent research has focused mainly on a very specific area, a fair comparison was not feasible.

6. Conclusion

In conclusion, we presented a new method to estimate image structure by finding distorted areas and reconciling curves within a framework. The result provides a structure which gives us the possibility of non-linear distortion correction. This method also provides a solution for high definition images, which are taken in rectangular shape. Moreover, the model can be used for videos without repetition on resampling and distortion estimation for each frame. Conducted multiple images and cropped images which contain unconventional features also are acceptable in our model.

References

[ABAO17] ANTUNES M., BARRETO J. P., AOUADA D., OTTERSTEN B.: Unsupervised vanishing point detection and camera calibration from a single manhattan image with radial distortion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 1

[AFAGSC14] ALEMÁN-FLORES M., ALVAREZ L., GOMEZ L., SANTANA-CEDRÉS D.: Line detection in images showing significant lens distortion and application to distortion correction. *Pattern Recognition Letters 36* (2014), 261–271. 2

[BD13] BUKHARI F., DAILEY M. N.: Automatic radial distortion estimation from a single image. *Journal of mathematical imaging and vision 45*, 1 (2013), 31–45. 2

[BERB18] BOGDAN O., ECKSTEIN V., RAMEAU F., BAZIN J.-C.: Deepcalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production* (2018), pp. 1–10. 2

[CFO93] CLEMENTINI E., FELICE P. D., OOSTEROM P. V.: A small set of formal topological relationships suitable for end-user interaction. In *International Symposium on Spatial Databases* (1993), Springer, pp. 277–295. 3

[CL19] CHEN J., LITTLE J. J.: Sports camera calibration via synthetic data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (2019), pp. 0–0. 2

[GCH*02] GIBSON S., COOK J., HOWARD T., HUBBOLD R., ORAM D.: Accurate camera calibration for off-line, video-based augmented reality. In *Proceedings. International Symposium on Mixed and Augmented Reality* (2002), IEEE, pp. 37–46. 2

[HGSE*18] HOLD-GEOFFROY Y., SUNKAVALLI K., EISENMANN J., FISHER M., GAMBARETTO E., HADAP S., LALONDE J.-F.: A perceptual measure for deep single image camera calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 2354–2363. 1

[HWZ*18] HUANG K., WANG Y., ZHOU Z., DING T., GAO S., MA Y.: Learning to parse wireframes in images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 626–635. 2

[HZZG20] HUANG K., ZIAUDDIN S., ZAND M., GREENSPAN M.: One shot radial distortion correction by direct linear transformation. In *2020 IEEE International Conference on Image Processing (ICIP)* (2020), IEEE, pp. 473–477. 1

[JAC*21] JEONG Y., AHN S., CHOY C., ANANDKUMAR A., CHO M., PARK J.: Self-calibrating neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5846–5854. 2

[KKL*20] KAKANI V., KIM H., LEE J., RYU C., KUMBHAM M.: Automatic distortion rectification of wide-angle images using outlier refinement for streamlining vision tasks. *Sensors 20*, 3 (2020), 894. 2

[LGL*21] LEE J., GO H., LEE H., CHO S., SUNG M., KIM J.: Ctrl-c: Camera calibration transformer with line-classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 16228–16237. 2

[LLC*21] LOCHMAN Y., LIEPIESHOV K., CHEN J., PERDOCH M., ZACH C., PRITTS J.: Babelcalib: A universal approach to calibrating central cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 15253–15262. 1

[LMG*19] LOPEZ M., MARI R., GARGALLO P., KUANG Y., GONZALEZ-JIMENEZ J., HARO G.: Deep single image camera calibration with radial distortion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 11817–11825. 2, 5

[LN18] LAURESHYN A., NILSSON M.: How accurately can we measure from video? practical considerations and enhancements of the camera calibration procedure. *Transportation Research Record 2672*, 43 (2018), 24–33. 2

[LZSL19] LI X., ZHANG B., SANDER P. V., LIAO J.: Blind geometric distortion correction on images through deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4855–4864. 1, 2, 4, 5

[PH95] PEI S.-C., HORNG J.-H.: Circular arc detection based on hough transform. *Pattern recognition letters 16*, 6 (1995), 615–625. 2

[RLZS14] RONG W., LI Z., ZHANG W., SUN L.: An improved canny edge detection algorithm. In *2014 IEEE international conference on mechatronics and automation* (2014), IEEE, pp. 577–582. 2

[SHF*20] SHA L., HOBBS J., FELSEN P., WEI X., LUCEY P., GANGULY S.: End-to-end camera calibration for broadcast videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 13627–13636. 2

[SZ15] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition, 2015. arXiv:1409.1556. 3

[TN11] TIAN Y., NARASIMHAN S. G.: Rectification and 3d reconstruction of curved document images. In *CVPR 2011* (2011), IEEE, pp. 377–384. 2

[WB09] WANG Z., BOVIK A. C.: Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine 26*, 1 (2009), 98–117. 4

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing 13*, 4 (2004), 600–612. 4

[WY21] WAKAI N., YAMASHITA T.: Deep single fisheye image camera calibration for over 180-degree projection of field of view. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 1174–1183. 2

[YZW20] YANG F., ZHAO Y., WANG X.: Camera calibration using projective invariants of sphere images. *IEEE Access 8* (2020), 28324–28336. doi:10.1109/ACCESS.2020.2972029. 1

[ZHZ*21] ZHAO K., HAN Q., ZHANG C.-B., XU J., CHENG M.-M.: Deep hough transform for semantic line detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). 2

[ZLK*17] ZHOU B., LAPEDRIZA A., KHOSLA A., OLIVA A., TORRALBA A.: Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017). 4